Proxy re-encryption schemes for IoT and crowd sensing

Full-Text version

Title: **Proxy Re-encryption Schemes for IoT and Crowd Sensing**

Authors: Daniel Díaz-Sánchez, *Senior Member, IEEE*
Telematic Engineering Department, Carlos III University, 28911, Leganés, Madrid, SPAIN
(e-mail: dds@it.uc3m.es)

R. Simon Sherratt, *Fellow, IEEE*
School of Systems Engineering, the University of Reading, RG6 6AY, UK
(e-mail: sherratt@ieee.org)

Patricia Arias, *Member, IEEE*
Telematic Engineering Department, Carlos III University, 28911, Leganés, Madrid, SPAIN
(e-mail: ariasp@it.uc3m.es)

Florina Almenares, *Member, IEEE*
Telematic Engineering Department, Carlos III University, 28911, Leganés, Madrid, SPAIN
(e-mail: florina@it.uc3m.es)

Andrés Marín López, *Member, IEEE*
Telematic Engineering Department, Carlos III University, 28911, Leganés, Madrid, SPAIN
(e-mail: amarin@it.uc3m.es)

Publication: **2016 IEEE International Conference on Consumer Electronics (ICCE)**
pp.: 15-16
Date: 7-11 Jan. 2016
Location: Las Vegas, Nevada, USA
DOI: 10.1109/ICCE.2016.7430505

Funding

This work has been partially funded by project INRISCO TEC2014-54335-C4-2-R and Jose Castillejo Mobility Grant CAS144/00364

Abstract

IoT, crowd sensing and smart cities will be a traffic challenge. New communication paradigms as asynchronous messaging carry and forward, scheduled delivery and temporary storage will be needed to manage network resources dynamically. Since traditional end to end security will require keeping security associations among devices for a long time draining valuable resources, we propose and evaluate the use of proxy re-encryption protocols in these scenarios as a solution for reliable and flexible security.

I. INTRODUCTION

The Internet of Things and other upcoming concepts as Smart Cities and Crowd Sensing depend on the available context information to deliver the appropriate quality. It can be considered the richest the information is, the better. Due to this, the purpose for devices to rely on the cloud is twofold: to process more information than they can on their own and to benefit from the information others push to the cloud.

These scenarios will influence the network as we know today. The information that flows to the cloud may be sensitive so it is important to set the appropriate security grounds to ensure only authorized recipients receive the data. Moreover, the huge amount of data currently flowing from devices to the cloud will increase in the near future whereas the outcome of that data processing, flowing from the cloud to the devices, will be smaller leading to an asymmetry. Finally, it should be considered that millions of devices sending data to several recipients over the network cannot be coordinated globally so alternative communication schemas should be considered to avoid collapsing the network. This entails considering asynchronous, unacknowledged, delayed delivery and carry-forward protocols. This work describes the use of proxy re-encryption protocols in these scenarios to handle asynchronous message driven communications as a solution.

II. RELATED WORK

The proxy re-encryption is based on the notion of "atomic proxy cryptography". It was first introduced in [1]. Basically, this technology proposes the use of a semi-trusted proxy that transforms a cipher text for Alice into a cipher text for Bob without actually accessing the plaintext. Popular well-known approaches to proxy re-encryption are BBS (Blaze, Bleumer & Strauss) [2] and AFGH (Ateniese, Fu, Green & Hohenberger) [3]. BBS is based on ElGamal. In BBS, the proxy, knowing the key for Alice to Bob direction, can derive the key for Bob to Alice direction as the multiplicative inverse. This is not a feature but a problem since the proxy that can be untrusted, does not need additional information to derive the inverse Moreover, to accomplish bidirectional communication two keys (direct and inverse) should be kept since deriving the inverse every time is computationally expensive.

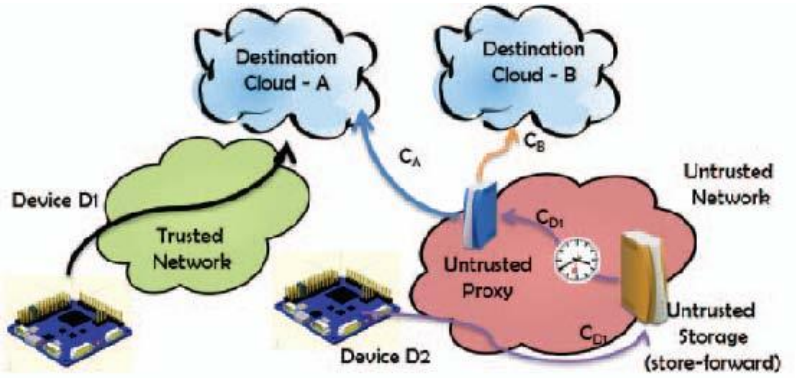


Fig. 1. Data from device D2 is stored in an untrusted cloud
and eventually delivered to Cloud A and B without compromising the data

AFGH is based on Bilinear maps and a proxy knowing the transformation key for one direction cannot derive the opposite direction key from it. Moreover, a message that has been encrypted for proxy re-encryption with AFGH can be recovered by the origin before and after proxy re-encryption permitting the source to recover, forward or re-encrypt a message for another destination if network conditions change again as it may happen in dynamic scenarios.

III. ARCHITECTURE AND OPERATION

We have modeled an IoT environment as a topology free system composed by different nodes. Every node can communicate with others directly or by means of intermediary nodes. Nodes can be used to model sensors, actuator, end user devices and intermediary systems ass communication gateways or M2M gateways. Moreover, nodes are also used to model the destination cloud nodes that collect the data for performing data analytics or business intelligence.

In order to characterize existing automation, sensor/actuator systems, nodes can be grouped in different topologies as line, mesh, tree or bus and connected too other devices using any topology through other nodes. When it comes to the security we have defined three different roles that can be assumed by any node in the system: endpoint nodes, key manager and re-encryption proxy. The endpoint nodes are nodes exchanging data trough untrusted networks. The key manager is the node acting as a key directory so a given source can find the destination public key using any addressing mechanism. The re-encryption proxy is the entity actually performing the re-encryption that may also group, store or delay messages according to a network policy or may also re-encrypt for alternative destinations if necessary.

Consider the scenario in Fig. 2. Devices in the trusted network communicate directly with the Destination Cloud A (CA) but upon a failure or a change in the network conditions that cloud becomes unreachable. Due to that the traffic should be re-routed through an untrusted network to a Destination Cloud B (CB) and will be eventually sent to CA.

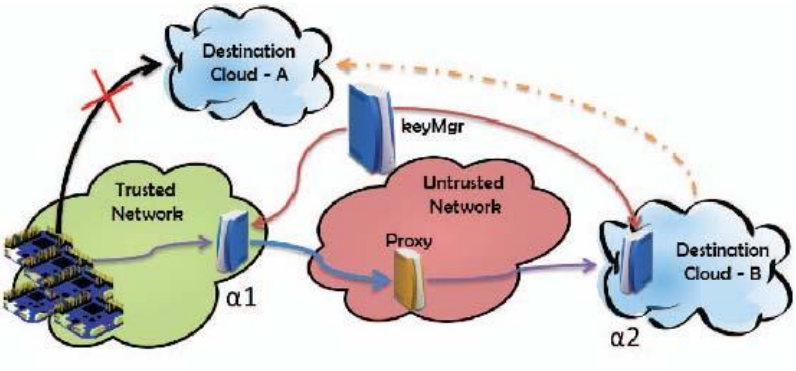


Fig. 2. Alternative delivery if data through an untrusted cloud upon a failure

For simplicity we consider all the traffic will be originated by $\alpha 1$ in the trusted network, processed by the proxy and re-encrypted for $\alpha 2$. Both $\alpha 1$ and $\alpha 2$ have a key pair. And thus both of them have calculated prime numbers q and r. $\alpha 1$ and $\alpha 2$ send their public key to key manager. $\alpha 1$ fetches the public key of $\alpha 2$ from the key manger and computes the re-encryption key ($RK_{\alpha 1 \to \alpha 2}$) as:

$$RK_{\alpha 1 \to \alpha 2} = {Pk_{\alpha 2}}^{1/Sk_{\alpha 1}} = g^{\frac{Sk_{\alpha 2}}{Sk_{\alpha 1}}} = g^{\frac{\alpha 2}{\alpha 1}} \quad (1)$$

Where $Sk_{\alpha 1} = \alpha$ is the private key and α a random integer belonging Zq; and $Pk_{\alpha 2} = g^{Sk_{\alpha 1}} = g^{\alpha}$ the public key. Once calculated, $\alpha 1$ sends $RK_{\alpha 1 \to \alpha 2}$ to the proxy. When $\alpha 1$ wants to send a message M and encrypt it to $\alpha 2$ it computer the cipher text $C_{1_{\alpha 1 \to \alpha 2}}$ as:

$$C_{1_{\alpha 1 \to \alpha 2}} = [A_1, B_1] = [g^{\alpha 1 \cdot r}, M \cdot Z^r] = [{PK_{\alpha 1}}^r, M \cdot Z^r] \quad (2)$$

$\alpha 1$ sends the encrypted message to the proxy. The proxy can store the messages to be delivered later, start sending the messages immediately or wait for any change in the destination (that would require a new RK to be computed). To illustrate the process, let's suppose the proxy re-encrypts the messages to the destination ($\alpha 2$) and delivers them. To do so it should compute $C_{2_{\alpha 1 \to \alpha 2}}$ as

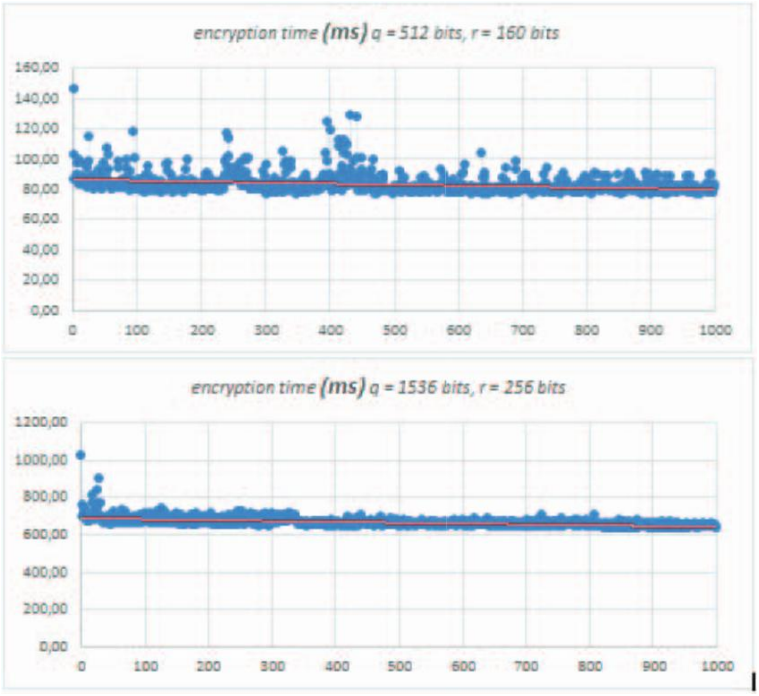
$$C_{2_{\alpha 1 \to \alpha 2}} = [A_2, B_2] = [e(A_1, RK_{\alpha 1 \to \alpha 2}), B_1] = \left[e\left(g^{\alpha 1 \cdot r}, g^{\frac{\alpha 2}{\alpha 1}}\right), M \cdot Z^r\right] = [Z^{r \cdot \alpha 2}, M \cdot Z^r] \quad (3)$$

Finally, $\alpha 2$ can acquire the original message M as follows:

$$\frac{B_2}{{A_2}^{1/Sk_{\alpha 2}}} = \frac{B_1}{{A_2}^{1/Sk_{\alpha 2}}} = \frac{M \cdot Z^r}{Z^{r \cdot \alpha 2/\alpha 2}} = \frac{M \cdot Z^r}{Z^r} = M \quad (4)$$

IV. Results and Conclusions

There are three important aspects for using proxy re-encryption in several scenarios under the scope of IoT, the first is the cost of the generation of the public and private key. There are several studies showing nowadays small processors are able to generate those parameters with no effort. The second important aspect to consider is the cost of generating the re-encryption key material since it may happen frequently in dynamic scenarios. Finally, the most worrying one is the cost of generating the encrypted message to enable proxy re-encryption. We have simulated several scenarios and collected data to analyze the impact of these three aspects. In this paper we show the cost of generating the encrypted messages can be prohibitive for long keys (q/r parameters) taking up to 25 seconds per messages (of the maximum size allowed by the key). However, shorter keys give and appropriate degree of privacy whereas keep the encrypted message generation time short enough to be useful for non-critical applications (and low traffic).



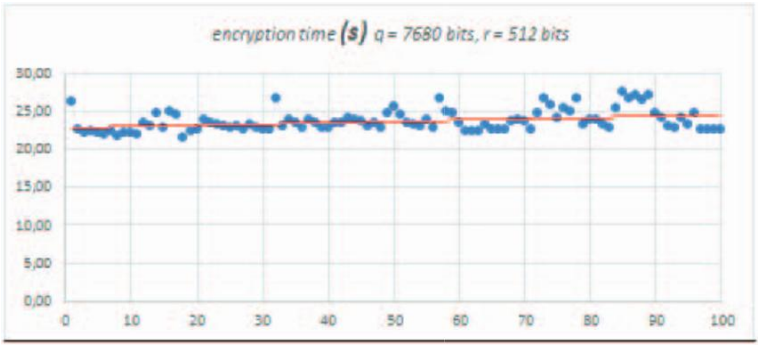


Fig. 3. Encryption time calculated 1000 times for different key lengths

We have stated these scenarios will prevail new challenges to future networks being paradigms as asynchronous messaging, carry and forward, scheduled delivery and temporary storage needed to manage network resources appropriately. Moreover, privacy will be critical for the adoption of new services. We have shown proxy re-encryption is an interesting tool for turning these scenarios into reality and we have tested the impact in these scenarios mimicking computing resources present in IoT devices. We finally came to the conclusion that a correct study on the necessary degree of security (key length) and the expected traffic will make proxy re-encryption usable and interesting for new services.

REFERENCES

[1] Masahiro Mambo and Eiji Okamotom, "Proxy Cryptosystems: Delegation of the Power to Decrypt Ciphertexts," IEICE Trans. Fund. Electronics Communications and Computer Science, E80-A/1:54–63, 1997.

[2] M. Blaze, G. Bleumer, and M. Strauss, "Divertible protocols and atomic proxy cryptography," In Proceedings of Eurocrypt ’98, vol. 1403, pp 127–144, 1998

[3] G. Ateniese, K. Fu, M. Green, S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," ACM Transactions on Information and System Security, vol. 9, pp 1-30, 2015.